Case-Based Capture and Reuse of Aerospace Design Rationale

Summary of Research

Principal Investigator: David B. Leake
http://www.cs.indiana.edu/~leake
leake@cs.indiana.edu

Period covered by report: 5/1/98–7/31/01

Institution:
Computer Science Department
Indiana University
Bloomington, Indiana, 47405

Award No NCC 2-1035

Statement on Subject Inventions: No subject inventions (37 CFR 401.14) were developed under this award.

1 Introduction

The goal of this project was to apply artificial intelligence techniques to facilitate capture and reuse of aerospace design rationale. The project combined case-based reasoning (CBR) (Kolodner 1993; Leake 1996; Watson 1997) and concept maps (CMaps) (Novak & Gowin 1984; Novak 1998) to develop methods for capturing, organizing, and interactively accessing records of experiences encapsulating the methods and rationale underlying expert aerospace design, in order to bring the captured knowledge to bear to support future reasoning. The project's results contribute both principles and methods for effective design-aiding systems that aid capture and access of useful design knowledge. This work has been conducted jointly with teams at Indiana University, directed by Dr. David Leake, and at the Institute for Human and Machine Cognition at the University of West Florida, directed by Dr. Alberto Cañas under a subcontract of this grant.

The project has been guided by the tenets that design-aiding systems must:

- Leverage a designer's knowledge, rather than attempting to replace it,
- Be able to reflect different designers' differing conceptualizations of the design task, and to clarify those conceptualizations to others,
- Include capabilities to capture information both by interactive knowledge modeling, and during normal use, and
- Integrate into normal designer tasks as naturally and unobtrusive as possible.

Throughout this project, the specific research and application areas pursued have been identified and refined through consultations with Mary Livingston, our technical contact at NASA Ames, and additional NASA aerospace design experts with whom she arranged contacts.

The project's research areas and results fall into four primary categories:

- **Case-Based Reasoning Support for Using Numerical Computation Software:** Results include principles and methods for the application of case-based reasoning to support large-scale numerical computation (e.g., as numerical simulations for aerospace design).
- **Concept Mapping Support for Flexible Knowledge Capture and Sharing:** Results include principles and methods for electronic *concept mapping* tools for knowledge modeling (e.g., to capture a designer's conceptualization of the factors relevant to a design decision). These have been implemented in a publicly-available set of electronic tools for concept mapping.
- **Proactive Case-Based Concept Map Retrieval:** Results include principles and methods for applying case-based reasoning methods to concept map retrieval, as well as empirical simulations studies on the factors affecting the performance of proactive context-based retrieval.

- **Context-Based Research Support:** Results include a prototype system to monitor information access and automatically suggest information resources that are relevant to the designer's information search (e.g.,when finding information to relevant to particular phases of spacecraft performance), as well as a competitive network algorithm for context identification.

The following report summarizes our research in each of these areas.

2 Case-Based Reasoning to Support Users of Numerical Computation Software

Numerical aerospace simulation requires the application of complex and specialized computational tools (e.g., computational fluid dynamics codes such as OVERFLOW). Developing effective solution strategies depends on making good choices about the selection, parameterization, and organization of component tools and resources. Because making good choices may require considerable effort and expertise, designing "intelligent" components that can make informed recommendations about solution development will play a valuable role in realizing the full potential of these tools and in facilitating their use. Case-based reasoning provides a promising framework for providing this support (Wilson, Leake, & Bramley 2000).

As the task domain for a pilot study, in consultation with Mary Livingston we selected the problem of acquiring the convergence criteria that experts use to guide the selection of useful data from a set of numerical simulations of high-lift systems. Initial research focused both on developing a case-based reasoning engine for this task, and on adapting and refining existing concept mapping tools to use to clarify and capture the rationale underlying designer experiences.

The first phase of this research developed and demonstrated a general-purpose CBR retrieval engine, augmented with the ability to manipulate two distinct types of cases: *Data cases,* encapsulating information about specific simulation runs, and *Strategy cases,* encapsulating textual information about the informal "engineering metrics" used by experts to determine convergence for practical problems. This reflects the fact that the choice of which engineering metrics to use depends on the task context. Storing information about the design task enables the user to inspect the link between convergence criteria and the circumstances under which they are selected, suggesting when particular criteria may be applicable in the future.

Our work on this pilot identified three types of research questions. The first concerned how to handle the many cases that a case-based support system would accumulate: the scale-up issues required to deal with large case bases, the case-base sharing issues as different users build up and exchange their own cases, and the maintenance of case information as new information became available. The second concerned providing a richer method for expressing the domain information needed to make high-quality choices. The third concerned how to provide effective domain-independent methods for determining task context. These became the focuses of our following work on maintaining and exploiting large distributed case bases, on concept mapping tools and intelligent support for resource access, and on automatic context extraction methods.

Case-Base Maintenance: Because case-base support systems will be required to manage substantial numbers of cases describing changing environments, we studied multiple issues in maintaining and compacting case bases. These include how to determine when to retain cases, given tradeoffs between increased retrieval costs with large case bases and performance limitations for small case bases; how to integrate case bases with database systems; and how to exploit case information from different repositories (Leake & Wilson 1998; 1999b; Sengupta, Wilson, & Leake 1999; Leake & Wilson 2000b; Wilson & Leake 2001).

Dealing with multiple case repositories is particularly important in component-based models, which provide the opportunity for multiple case bases, each associated with a particular component, but also raise issues of case-base standardization and applying cases generated in one context to support problem-solving in another. For example, if CBR methods were used to predict problem-solving time, the machine on which cases were generated would have a strong effect on predictions. However, with appropriate corrections for systematic differences, case bases from different users can still provide a valuable resource to guide future decision-making. (Leake & Sooriamurthi 2001) presents an examination of the tradeoffs between relying on smaller, more task-specific case bases compared to performing retrievals from a wider range of cases from somewhat different, but related tasks. Those results have shown the potential benefits of drawing on external case bases, especially early in the development of a task-specific case base, and support the practicality of domain-independent "weak methods," combined with ensemble techniques, for automatically adapting results across tasks.

3 Concept Mapping Support for Flexible Knowledge Capture and Sharing

Capturing expert knowledge is an essential component of the knowledge management process. Once models of experts' domain knowledge are available, they can provide a valuable resource for knowledge comparison, refinement, and reuse. However, a difficult question is how to obtain the required knowledge models. Hand-crafting them is expensive; using machine learning techniques may not be effective. We have investigated an alternative approach: developing tools to allow experts themselves to construct models of their knowledge. Our approach builds on *concept mapping* (Novak & Gowin 1984), in which experts themselves construct a two-dimensional, visually-based representation of concepts and their relationships, using the terms and knowledge organization that are most natural to them. Concept mapping provides a flexible method for capturing, elucidating, and sharing aerospace knowledge and rationale.

Concept mapping was first proposed in educational settings, to help assess students' understanding and to aid them in refining their understanding by making it explicit and comparing it with the understanding of others. In the concept mapping view, experts who build concept maps are not simply externalizing pre-existing internal knowledge, but are also doing knowledge construction. Thus intelligent tools to provide relevant knowledge for them to consider and compare during concept mapping could facilitate not only knowledge capture, but knowledge generation as they augment their understanding.

With the support of this grant, the Institute for Human and Machine Cognition of the University of West Florida has been pursuing research and development into a set of general tools to enable the construction of navigable models based on concept maps, over the Internet or locally on hard disk or CD-ROM. These tools include a concept-map based browser that allows the user to navigate through concept map-based multimedia environments and case libraries, transparently navigating through local or remote servers, accessing media anywhere on the Internet. The server is designed for scalability, allowing a large number of connections accessing many models with extensive collections of media. The system is written in Java to be platform-independent and to facilitate its integration with the CBR system and existing design tools.

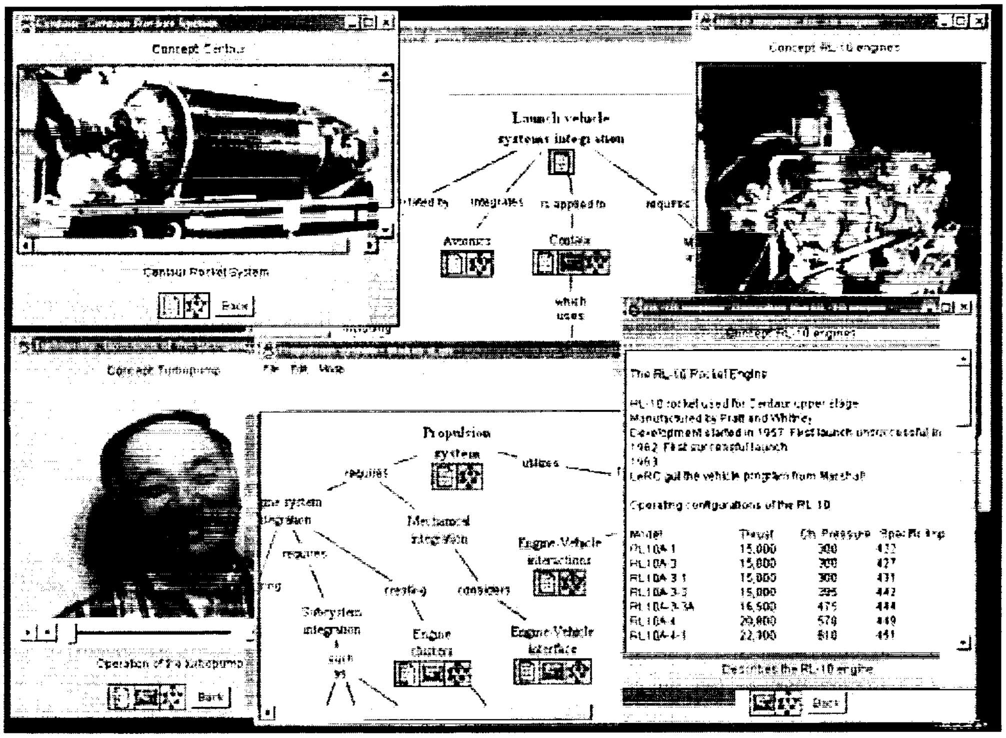
Figure 1 shows a screen shot from these tools, capturing information about the design of the Centaur rocket. The center of the image shows concept maps describing the launch vehicle systems integration and rationale for the design of the propulsion system, with nodes for concepts connected with named links expressing concept relationships. Concept maps appear similar to semantic nets but have no fixed semantics and vocabulary—they simply make explicit any set of concepts and relationships in any vocabulary that the expert chooses.

As also shown in the figure, electronic concept maps enable associating concept map nodes with other resources, such as photographs, video clips, and textual passages, as well as pointers to additional concept maps, in order to define a hierarchical concept structure. The result is a rich and flexible concept representation to help humans understand domains and revise their own knowledge.

Results on the concept map system include:

- Development of browser tools to allow the user to transparently navigate through local or remote concept map servers, accessing media anywhere on the Internet. and
- Development of software capabilities necessary to run a scalable server that will store concept maps and the corresponding media. Particularly relevant in the design of this software is that the server has been developed to allow large number of connections accessing many models with extensive collections of media.

During the first and second years of the project, most of the development effort concentrated on finishing the major functionality of the "Core," a modular plug-in architecture for



The RL-10 Rocket Engine

RL-10 rocket used for Centaur upper stage
Manufactured by Pratt and Whitney
Development started in 1957. First launch unsuccessful in 1962. First successful launch
1963
LeRC got the vehicle program from Marshall

Operating configurations of the RL-10

Model	Thrust	Ch. Pressure	Specific Imp.
RL10A-1	15,000	300	422
RL10A-3	15,000	300	427
RL10A-3-1	15,000	300	431
RL10A-3-3	15,000	395	442
RL10A-3-3A	16,500	475	444
RL10A-4	20,800	578	449
RL10A-4-1	22,300	610	451

Figure 1: A screen shot of a concept map and associated multimedia resources on the design of the Centaur rocket, prepared and displayed using the University of West Florida concept mapping tools.

the concept mapping tools. The Core allows constructing applications by first decomposing the application into individual pieces, implementing each piece as a module, and then constructing the application by assembling the required set of modules.

Completion of the core allowed us to develop the module to interface with the case-based reasoning components, enabling case-based reasoning software to interact with the concept map application and vice-versa, allowing the capture and examination of rationale information in both forms. Additional refinements include improvements to functionality such as the ability to export concept maps as graphics files for viewing on the web, and new user profile and security capabilities.

The concept mapping tools are now robust beta versions that can be downloaded for Windows, Macintosh, and Solaris; they are accompanied by system documentation and tutorials on concept map generation, all of which are available from http://cmap.coginst.uwf.edu/.

4 Proactive Case-Based Concept Map Retrieval

Making effective use of concept maps depends on being able to access the right concept maps at the right time. Likewise, the knowledge modeling process may be facilitated by consulting relevant concept maps from similar situations. To explore how to provide this support and the tradeoffs it involved, we developed of a testbed proactive design support tool, DRAMA (Design Retrieval and Adaptation Mechanisms for Aerospace), that combines case-based reasoning and concept mapping tools to aid designers as they retrieve, consult, and revise prior designs represented in the form of concept maps.

DRAMA's Integrated Interactive Design Support Approach: Ideally, case-based design support tools will include three related capabilities to aid design reuse: capture of and access to specific design experiences, support for new designers as they try to understand the lessons of those prior experiences; and support for adapting prior designs to fit new design goals. For practical application, the tools must not depend on extensive domain knowledge; for designer acceptance, they must leave the designer in control. To investigate methods for providing case-based support and knowledge capture, in the second funding period we developed a testbed design aiding system that integrates case-based reasoning with interactive tools for capturing expert design knowledge through concept mapping, with the goal of leveraging off the strengths of both approaches. The concept mapping tools provided an interactive interface and crucial functions for generating and examining design cases, as well as navigating their hierarchical structure, and case-based reasoning provided the ability to automatically retrieve information about similar designs. Figure 2 illustrates the types of information manipulated by the system.

DRAMA supports browsing of prior design knowledge and proactively provides designers with concrete examples of designs and design adaptations from similar prior problems. At the same time, it unobtrusively acquires new examples from the user's interactive design process

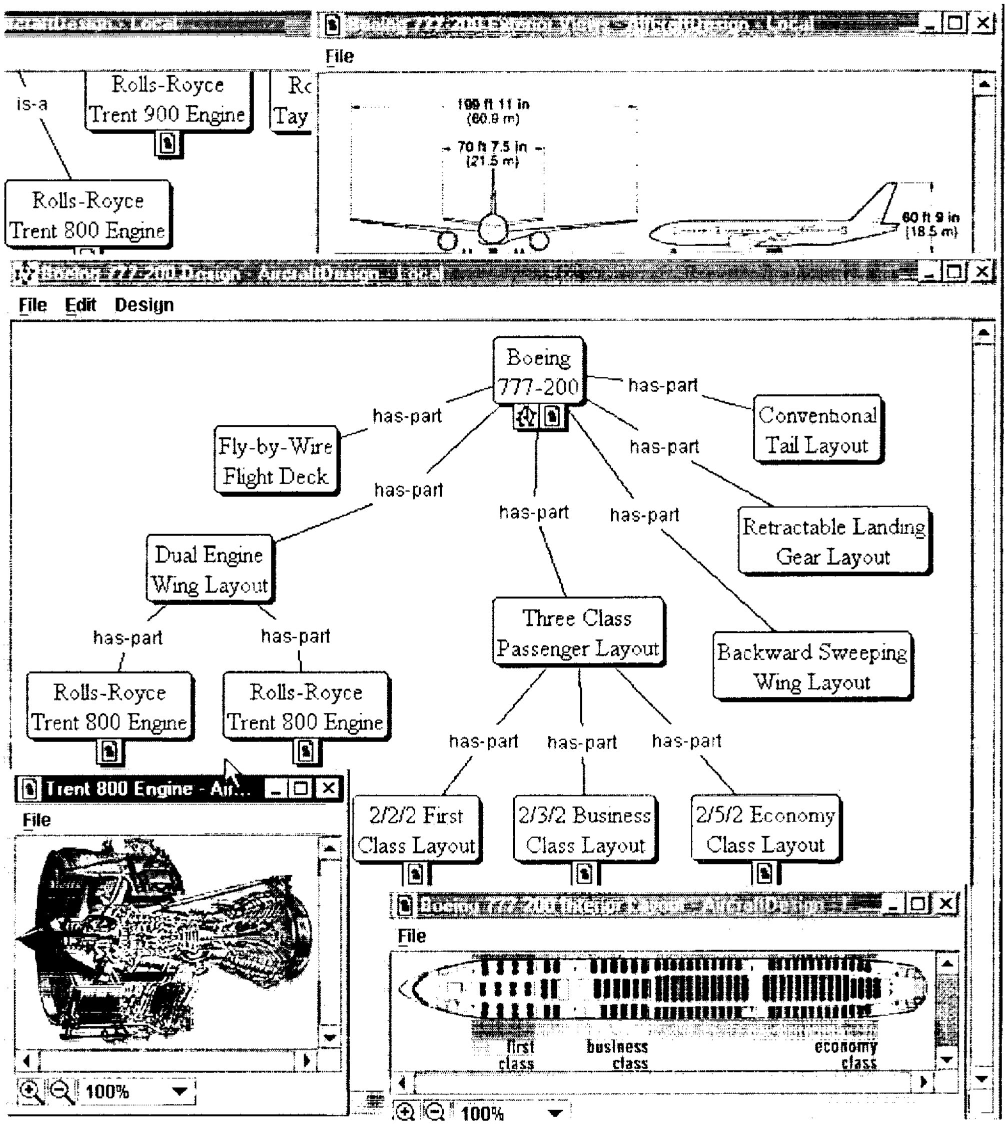


Figure 2: A screen shot from the DRAMA system, showing the integration of multiple aircraft design resources through concept maps.

(Cañas, Leake, & Wilson 1999; Leake & Wilson 1999a; 2000). This work advanced the state of the art in case-based reasoning in allowing flexible, user-defined and user-changeable case representations.

We evaluated the DRAMA concept with a set of simulation experiments testing the system's ability to dynamically adjust the relevance criteria for retrieving prior experiences, exploiting task-based information that is captured automatically, without requiring the user to provide it explicitly. In (Leake & Wilson 2000) we present experiments examining this behavior in detail, demonstrating how the benefits of this automatic retrieval approach vary under conditions modeling different levels of design novelty, different stages in the design process, and different levels of user expertise.

Domain-independent concept map retrieval algorithms: Building on our experience with DRAMA in addressing the similarity of concept maps, we developed domain-independent algorithms for performing concept map retrieval (Cañas, Leake, & Maguitman 2001). These methods combine aspects of information retrieval and topological analysis (Kleinberg 1999) to summarize both the content and structure of concept maps, in order to retrieve similar prior concept maps and related resources.

5 Context-Based Research Support

Expert designers acquire considerable knowledge about relevant sources of information, and develop rich domain models. However, our NASA contacts observed that serious problems arise as designers move into new areas (e.g., from aircraft design to the design of space vehicles). In this situation, designers may have to expend considerable effort to master the new areas and find the answers they need. Consequently, systems to help their research process and profit from others' research experience could play a significant role in facilitating the transition to new areas and projects. Thus our final main focus has been to develop case-based methods to aid the designer's learning, knowledge capture, and reuse when designers must deal with novel problems that require additional research (e.g., to decide the feasibility of a particular wing shape, or whether performance or cost would be increased by including a particular feature in the design of a wing).

We have developed and implemented a system, CALVIN (Context Accumulation for Learning Varied Informative Notes) (Leake *et al.* 2000), which investigates how case-based systems can support the process of finding information relevant to a particular task. The user of such a research-aiding system may have any of a wide range of domain-level tasks, such as finding the rationale underlying a decision, deciding the likely outcomes of different courses of action, generating feasibility studies, generating or refining designs, etc. These tasks give rise to information needs, which the user must satisfy through research. CALVIN captures lessons about where and how to find information relevant to the user's task. These lessons are used to provide future users with suggestions of relevant information resources (e.g., web pages, documents, people who have researched similar problems, etc.)

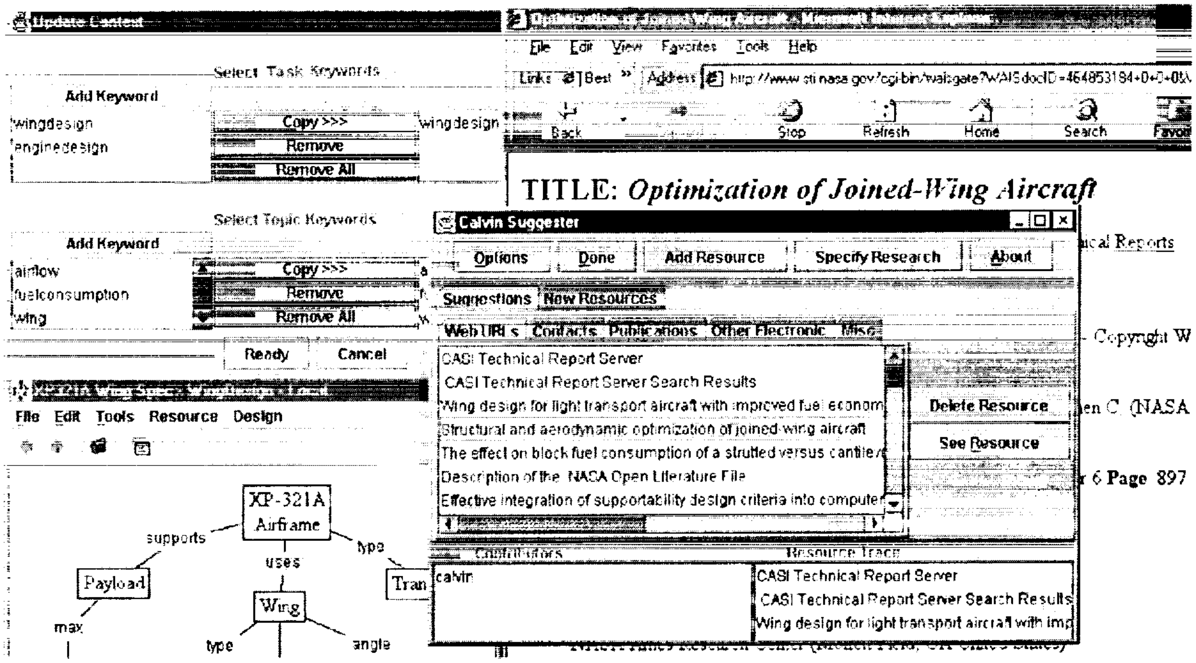
CALVIN is "resource suggester" system for capturing and proactively providing task-relevant information, based on monitoring a user's research process. The source suggester learns task-driven lessons and proactively provides them in similar situations, to aid in navigation of heterogeneous information sources. Our design draws on collaborative work with the Northwestern University InfoLab on the requirements for proactive design-support systems (Leake *et al.* 1999b; 1999a) and on methods for selecting the specialized information sources likely to be useful for finding particular types of information (Leake *et al.* 1999c).

CALVIN's cases suggesting information sources are complementary to cases containing lessons concerning domain-level issues, and both types of knowledge can play important roles in aiding a designer. CALVIN includes the capability for users to add textual annotations to resources describing important points, and uses concept mapping tools to provide a sketchpad for elucidating the user's growing knowledge of the domain as research progresses. These concept maps provide a supplementary type of information resource for CALVIN to suggest in similar contexts, and a medium for designers both to clarify their own understanding and to elucidate it for sharing rationale. The CALVIN system is designed to run in parallel with the concept mapping system; users can use concept maps to maintain ongoing traces of the understanding grained during their research and can their record concept maps as resources to be suggested automatically for browsing when future designers study similar areas. Thus CALVIN stores both lessons about how to learn information, and records of the domain lessons learned.

In principle, the research task supported by CALVIN could involve a number of different software tools, such as browsers, search engines, databases, electronic mail, etc., as well as telephone or face-to-face contact. The current system directly monitors only web browsers, but provides capabilities for users to record information about other searches, or resources, and their results (for example, descriptions of books or articles that are available off-line, or about research paths taken to find useful contact people). This information is stored in "resource cases," which are indexed by the task and current search context to be retrieved automatically in similar future contexts.

As the user browses pages using a standard browser, the system stores cases representing the content of the pages and the contexts in which they were considered. The system generates a simple description of the page content (without natural language processing) by extracting keywords from the pages visited and combining those keywords with the previously-generated context to suggest resources that were useful in similar prior circumstances. A screen image including the current Suggester display is shown in figure 3.

As part of the investigation of the research support process, we have developed methods for selecting task-relevant information sources (Leake & Scherle 2001) and have developed WordSieve, an algorithm for automatically extracting information about the context in which documents are consulted during web browsing (Bauer & Leake 2001b). Using



wingdesign
enginedesign

wingdesign

airflow
fuelconsumption
wing

http://www.sti.nasa.gov/cgi-bin/waisgate?WAISdocID=464853184+0+0+0&

TITLE: *Optimization of Joined-Wing Aircraft*

CASI Technical Report Server
CASI Technical Report Server Search Results
Wing design for light transport aircraft with improved fuel econom
Structural and aerodynamic optimization of joined-wing aircraft
The effect on block fuel consumption of a strutted versus cantilev
Description of the NASA Open Literature File
Effective integration of supportability design criteria into computer

calvin

CASI Technical Report Server
CASI Technical Report Server Search Results
Wing design for light transport aircraft with imp

Figure 3: Windows from a CALVIN run: CALVIN's suggester window and context update window, with windows for a suggested web page and related concept map.

information extracted from the stream of documents consulted by the user, WordSieve automatically builds context profiles which differentiate sets of documents that users tend to access in groups. These profiles are used in a research-aiding system to index documents consulted in the current context and pro-actively suggest them to users in similar future contexts. In initial experiments on the capability to match documents to the task contexts in which they were consulted, WordSieve indexing outperformed indexing based on *Term Frequency/Inverse Document Frequency*, a common document indexing approach for intelligent agents in information retrieval. We are continuing this work to perform evaluations against benchmark retrieval methods. In conjunction with this work we have developed a distributed agent-based architecture for research support (Bauer & Leake 2001a).

6 Summary

The previous sections summarize the key task areas studied under the funding of this grant and the research progress in each one. In particular, the funded research has provided advances in case-based methods for capture and reuse of aerospace design rationale in four main areas:

- Case-based reasoning support for using numerical computation software
- Concept mapping support for flexible knowledge capture and sharing

- Proactive case-based concept map retrieval
- Context-based research support

Beyond the applicability of the project results to aerospace design support, the results of this project have wide applicability to other types of tasks. Full results on the advances in research principles are available in the papers cited in this summary, and many of the advances in concept mapping tools described here are now implemented in the publicly available concept mapping tools that may be downloaded from the Institute for Human and Machine Cognition (http://cmap.coginst.uwf.edu/). In addition, the research results presented here have provided a foundation for a proactive, case-based concept map retrieval module that we expect to be included in the next release of these tools.

References

Bauer, T., and Leake, D. 2001a. A research agent architecture for real time data collection and analysis. In *Working Notes of the Second International Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems.* New York: ACM. Fifth International Conference on Autonomous Agents.

Bauer, T., and Leake, D. 2001b. Wordsieve: A method for real-time context extraction. In *Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, Context 2001.* Berlin: Springer-Verlag.

Cañas, A.; Leake, D.; and Maguitman, A. 2001. Combining concept mapping with cbr: Towards experience-based support for knowledge modeling. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference.* Menlo Park: AAAI Press.

Cañas, A.; Leake, D.; and Wilson, D. 1999. Managing, mapping, and manipulating conceptual knowledge. In *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning*, 10–14. Menlo Park: AAAI Press.

Kleinberg, J. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46(5):604–632.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

Leake, D., and Scherle, R. 2001. Towards context-based search engine selection. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, 109–112.

Leake, D., and Sooriamurthi, R. 2001. When two case bases are better than one: Exploiting multiple case bases. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01.* Berlin: Springer-Verlag.

Leake, D., and Wilson, D. 1998. Case-base maintenance: Dimensions and directions. In Cunningham, P.; Smyth, B.; and Keane, M., eds., *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, 196–207. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 1999a. Combining CBR with interactive knowledge acquisition, manipulation and reuse. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 203 217. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 1999b. When experience is wrong: Examining CBR for changing tasks and environments. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 218 232. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 2000a. A case-based framework for interactive capture and reuse of design knowledge. *Applied Intelligence* 14. In press.

Leake, D., and Wilson, D. 2000b. Remembering why to remember: Performance-guided case-base maintenance. In Blanzieri, E., and Portinale, L., eds., *Proceedings of the Fifth European Workshop on Case-Based Reasoning*, 161-172. Berlin: Springer Verlag.

Leake, D.; Birnbaum, L.; Hammond, K.; Marlow, C.; and Yang, H. 1999a. Integrating diverse information resources in a case-based design environment. *Engineering Applications of Artificial Intelligence* 12(6):705-716.

Leake, D.; Birnbaum, L.; Hammond, K.; Marlow, C.; and Yang, H. 1999b. Task-based knowledge management. In *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning*, 35-39. AAAI Press.

Leake, D.; Scherle, R.; Budzik, J.; and Hammond, K. 1999c. Selecting task-relevant sources for just-in-time retrieval. In *Proceedings of the AAAI-99 Workshop on Intelligent Information Systems*, 38 43. Menlo Park: AAAI Press.

Leake, D.; Bauer, T.; Maguitman, A.; and Wilson, D. 2000. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In Aha, D., and Weber, R., eds., *Proceedings of the AAAI-2000 Workshop on Intelligent Lessons Learned Systems.* Menlo Park: AAAI Press.

Leake, D., ed. 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions.* Menlo Park, CA: AAAI Press/MIT Press.

Novak, J., and Gowin, D. 1984. *Learning How to Learn.* New York: Cambridge University Press.

Novak, J. 1998. *Learning, Creating, and Using Knowledge.* Mahwah, New Jersey: Erlbaum.

Sengupta, A.; Wilson, D.; and Leake, D. 1999. On constructing the right sort of CBR implementation. In *Proceedings of the IJCAI-99 Workshop on Automating the Construction of Case Based Reasoners.*

Watson, I. 1997. *Applying Case-Based Reasoning: Techniques for Enterprise Systems.* San Mateo, CA: Morgan Kaufmann.

Wilson, D., and Leake, D. 2001. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence* 17(2):196-213.

Wilson, D.; Leake, D.; and Bramley, R. 2000. Case-based recommender components for scientific problem-solving environments. In *Proceedings of the Sixteenth IMACS World Congress.* CD-ROM, Session 105, Paper 2.

7 Publications Acknowledging the Grant

Copies of most of these publications are available on-line at
http://www.cs.indiana.edu/˜leake/INDEX.html.

References

Bauer, T., and Leake, D. 2001a. A research agent architecture for real time data collection and analysis. In *Working Notes of the Second International Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems.* New York: ACM. Fifth International Conference on Autonomous Agents.

Bauer, T., and Leake, D. 2001b. Wordsieve: A method for real-time context extraction. In *Modeling and Using Context: Proceedings of the Third International and Interdisciplinary Conference, Context 2001.* Berlin: Springer-Verlag.

Cañas, A.; Leake, D.; and Maguitman, A. 2001. Combining concept mapping with cbr: Towards experience-based support for knowledge modeling. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference.* Menlo Park: AAAI Press.

Cañas, A.; Leake, D.; and Wilson, D. 1999. Managing, mapping, and manipulating conceptual knowledge. In *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning*, 10–14. Menlo Park: AAAI Press.

Leake, D., and Scherle, R. 2001. Towards context-based search engine selection. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, 109–112.

Leake, D., and Sooriamurthi, R. 2001. When two case bases are better than one: Exploiting multiple case bases. In *Case-Based Reasoning Research and Development: Proceedings of the Fourth International Conference on Case-Based Reasoning, ICCBR-01.* Berlin: Springer-Verlag.

Leake, D., and Wilson, D. 1998. Case-base maintenance: Dimensions and directions. In Cunningham, P.; Smyth, B.; and Keane, M., eds., *Proceedings of the Fourth European Workshop on Case-Based Reasoning*, 196–207. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 1999a. Combining CBR with interactive knowledge acquisition, manipulation and reuse. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 203–217. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 1999b. When experience is wrong: Examining CBR for changing tasks and environments. In *Proceedings of the Third International Conference on Case-Based Reasoning*, 218–232. Berlin: Springer Verlag.

Leake, D., and Wilson, D. 2000. A case-based framework for interactive capture and reuse of design knowledge. *Applied Intelligence* 14. In press.

Leake, D., and Wilson, D. 2000b. Remembering why to remember: Performance-guided case-base maintenance. In Blanzieri, E., and Portinale, L., eds., *Proceedings of the Fifth European Workshop on Case-Based Reasoning*, 161–172. Berlin: Springer Verlag.

Leake, D.; Birnbaum, L.; Hammond, K.; Marlow, C.; and Yang, H. 1999a. Integrating diverse information resources in a case-based design environment. *Engineering Applications of Artificial Intelligence* 12(6):705–716.

Leake, D.; Birnbaum, L.; Hammond, K.; Marlow, C.; and Yang, H. 1999b. Task-based knowledge management. In *Proceedings of the AAAI-99 Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning*, 35–39. AAAI Press.

Leake, D.; Scherle, R.; Budzik, J.; and Hammond, K. 1999c. Selecting task-relevant sources for just-in-time retrieval. In *Proceedings of the AAAI-99 Workshop on Intelligent Information Systems*, 38–43. Menlo Park: AAAI Press.

Leake, D.; Bauer, T.; Maguitman, A.; and Wilson, D. 2000. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In Aha, D., and Weber, R., eds., *Proceedings of the AAAI-2000 Workshop on Intelligent Lessons Learned Systems.* Menlo Park: AAAI Press.

Sengupta, A.; Wilson, D.; and Leake, D. 1999. On constructing the right sort of CBR implementation. In *Proceedings of the IJCAI-99 Workshop on Automating the Construction of Case Based Reasoners.*

Wilson, D., and Leake, D. 2001. Maintaining case-based reasoners: Dimensions and directions. *Computational Intelligence* 17(2):196–213.

Wilson, D.; Leake, D.; and Bramley, R. 2000. Case-based recommender components for scientific problem-solving environments. In *Proceedings of the Sixteenth IMACS World Congress.* CD-ROM, Session 105, Paper 2.